DOI: 10.3745/KIPSTC.2008.15-C.6.439

EPC 시스템간의 인증 및 암호화 방법의 설계 및 구현

김 대 중[†] · 김 정 재[††] · 이 승 민[†††] · 전 문 석[††††]

요 약

최근 EPCglobal Network의 중요성과 RFID 기술 및 응용에 대해 활발한 연구가 계속해서 증가하고 있으며, 유통 및 물류 분야뿐 아니라 각 산업 분야에서 다양한 응용 시스템들이 제안되고 있다. 현재 표준안이 제정되고 있는 EPCglobal Network에서 암호화 방법에 대해서는 X.509를 이용한다는 것만 명시되었으며, 정확한 스펙에 대해서는 아직 나오지 않은 상황이다. 따라서 본 논문에서는 EPCglobal Network를 구성하는 웹 서비스에 대해 메시지를 안전하게 전달하고, 보내온 메시지가 유효한 메시지인지 검증하는 방법, 안전하게 전달하기 위한 암호화 방법, 시스템간의 인증기법을 제안하고 구현한다.

키워드 : EPCglobal Network, EPC, RFID, PKI, 암호화, 시스템간 인증

Design and Implementation of Method of Authentication and Cryptography between EPC Systems

Kim Dae Jung[†] · Kim Jung Jae[††] · Lee Seung Min[†††] · Jun Moon Seog[††††]

ABSTRACT

Recently, the importance of EPCglobal Network and brisk researches on the RFID technologies and application have been increasing, also a number of industries including distribution and logistics are proposing various systems of the application. The Standard for EPCglobal Network, as now being in the process of its legislation, stipulates X.509 only for the method of encryption, without accurate specifications. This paper is, thus, to suggest the way of sending safely messages for the web-based service constituting EPCglobal Network, of verifying whether the received messages are effective, of encoding the messages for safer sending and of certifying between systems, and then to implement the way.

Keywords : EPCglobal Network, EPC, RFID, PKI, Symmetric Key, Asymmetric Key

1. 서 론

EPCglobal 아키텍처 프레임워크는 Electronic Product Code(EPC)를 사용하여 공급/유통망 강화라는 공통 목표를 위해 서비스되는 모든 것, EPCglobal과 위임기관이 운용하는 코어 서비스(EPCglobal Core Service)와 데이터 인터페이스, 소프트웨어, 하드웨어의 관련 표준(EPCglobal Standard)의 종합으로 EPCglobal 가입자가 EPCglobal과 EPCglobal 아키텍처 프레임워크 요소를 사용하는 개별 가입자와 상호작용하는 시너지 효과를 비공식으로 "EPCglobal Network"라고도 한다. 최근 이와 같은 EPCglobal Network의 중요성과 RFID(Radio Frequency IDentification) 기술 및 응용에 대해 활발한 연구가 이루어지고 있으며, 유통 및 물류 분야뿐 아니라 각 산업 분야에서 다양한 응용 시스템들이 제안되고 있다[1]. 현재 X.509 기반의 인증서 메시지 내용을 정의하고 이를 통해 모든 메시지를 암호화하고, 전자서명을 통해 데이터의 무결성을 유지하고, 데이터전송은 SOAP을 통해 전송하는 방안만 나와 있을 뿐 아직 정확히 CA는 어디에 존재시키며, 어떤 메시지를 암호화 할 것인지에 대해서는 정해지지 않았다. 본 논문에서는 EPCglobal Network를 구성하는 웹서비스에 대해 메시지를 안전하게 전달하고, 보내온 메시지가 유효한 메시지인지 검증하는 방법, 안전하게 전달하기 위한 암호화 방법, 시스템간의 인증기법을 제안하고 구현한다. 단순히 X.509 기반의 공개키 암호화 알고리즘을 사용한 방법보다 대칭키 암호화 알고리즘을 혼용하여 사용함으로 인해 메시지의 암호화 복호화 속도가 개선되었으며, 시스템간의 오버헤드 역시 감소되는 것을 확인하였다.

† 정 회 원 : 상호저축은행중앙회 IT본부 전산기획팀장(교신저자)
†† 정 회 원 : (주)리테일테크 기술연구소 수석연구원
††† 준 회 원 : 숭실대학교 컴퓨터학과 박사과정
†††† 종신회원 : 숭실대학교 컴퓨터학부 부교수

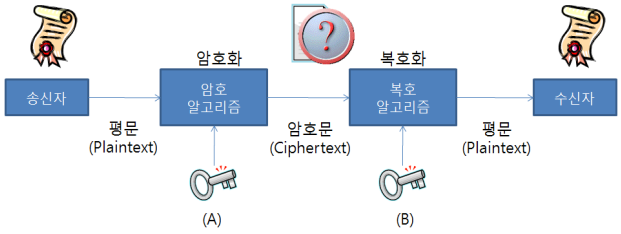
논문접수 : 2007년 10월 25일
수 정 일 : 1차 2008년 1월 26일, 2차 2008년 2월 26일
심사완료 : 2008년 3월 17일

본 논문의 구성은 다음과 같다. 2절은 관련연구로서 EPCglobal Network에 대한 전반적인 개념 및 암호화 방법을 소개하고, 3절에서는 제안하는 시스템 암호화 방법, 4절에서는 구현 방법, 5절에서는 실험평가, 6절에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 암호화 알고리즘

송신자는 (그림 1)과 같이 전송하고자 하는 평문을 암호화키(A)와 암호 알고리즘을 통해 암호문으로 변환하여 수신자에게 전송하고, 수신자는 복호화키(B)와 복호 알고리즘을 사용하여 원래의 평문을 생성한다.



(그림 1) 암호 및 복호화

2.2 대칭키 암호화 알고리즘

대칭키 암호 알고리즘(Symmetric Crypto Algorithm)은 데이터를 암호화하는 암호화키(A)와 암호문을 원래의 데이터로 바꿔주는 복호화키(B)가 같은 알고리즘을 의미한다. 따라서 통신에서 대칭키 암호 알고리즘을 사용하기 위해서는 통신하고자 하는 쌍방이 사전에 암호화키를 서로 공유해야만 한다. 이러한 성질을 따라 대칭키 암호 알고리즘은 대칭 암호 알고리즘이라고 한다.

데이터를 주고받는 경우 암호화하는 사람과 복호화 하는 사람이 다르다면 두 사람 모두 암호화와 복호화에 쓰이는 키를 알고 있어야 하므로, 암호화에 쓰인 키를 또 다른 정보보호 기술로 상대방에게 어떤 방법으로 전달하는 것이 문제가 된다. 하나의 통신망에 가입자 수가 n이라면 상호 교환해야 할 키의 개수는 ${}_nC_2 = \frac{n(n-1)}{2}$ 개로써 인터넷과 같은 불특정 다수를 대상으로 할 때에는 키 관리 방법이 어렵다는 단점도 있다[2].

2.3 공개키 암호화 알고리즘

대칭키 암호화 알고리즘에서의 문제점인 키 교환 문제점을 해결해 줄 수 있는 암호화 방식이 공개키 암호화 알고리즘(Public-Key Crypto Algorithm)이다. 공개키 암호 알고리즘은 (그림 1)과 같이 암호화할 때 쓰이는 키(A)와 복호화하는데 쓰이는 키(B)가 서로 다른 키가 존재하는 알고리즘으로 비대칭 암호 알고리즘(Asymmetric Crypto Algorithm)이라고도 한다. 두 개의 키는 수학적 함수를 기반으로 서로 연관이 있는 키 쌍으로 하나의 키는 누구든지 사용할 수 있게 공개하고 다른 하나는 외부에 공개되지 않도록 보관한다. 이때 공개하는 키를 공개키(Public Key)라고 하며 자신만이 볼 수 있도록 보관하는 키를 개인키(Private Key)라고 한다.

개인키로 메시지를 암호화하여 보내게 되면, 송신자의 공개키를 가지고 있는 사람은 누구나 그 내용을 확인할 수 있게 되지만, 수신자는 보낸 메시지가 송신자 본인이라는 것은 누구도 부인할 수가 없게 된다. 이유는 오직 송신자만이 그 자신의 개인키를 알고 있기 때문이며, 이 방법은 서명과 같은 것으로 전송 도중에 제3자의 개입여부를 수신자는 쉽게 알 수가 있는 방법이다[3]. 하지만 키 사이즈가 크다는 것과 2진수를 10진수로 변환하는 연산시간이 길다는 단점이 있다.

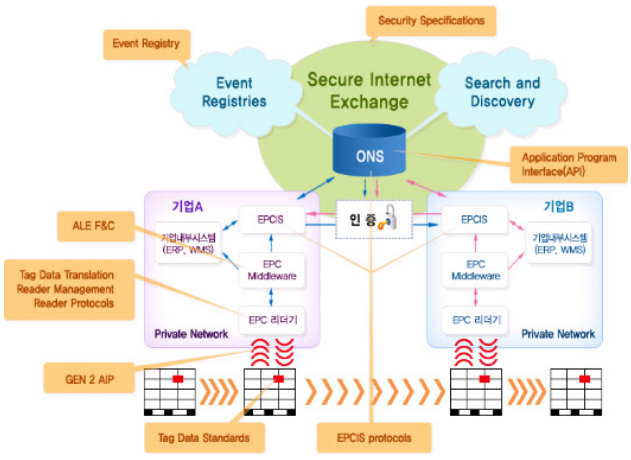
2.4 EPCglobal Network

EPCglobal Network란 EPC 코드와 RFID 기술을 바탕으로 하여 제품에 식별번호를 부여하고 정보를 저장할 수 있는 공간을 네트워크로 연동하여 공급자, 수요자, 그리고 소비자가 제품에 관련된 정보와 제품에 관련된 정보를 알 수 있게 해주는 시스템을 말한다[4].

EPCglobal은 (그림 2)와 같이 구성된다.

① EPC(Electronic Product Code)
EPC는 MIT Auto-ID 센터에서 개발된 코드체계로 물리적 또는 가상적으로 존재하는 물품 또는 서비스에 고유한 일련번호를 부여하여 식별을 가능하게 해주는 코드이다[5].

② RFID Tag (Gen2 : Class 1 Generation 2 UHF Air Interface Protocol)
2006년 5월 ISO에서 EPCglobal가 제안한 Gen2를 최종 승인하였으며, 제조사에서 물품을 제조할 때 각 물품 및 박스, 팔레트 등 제품이 이동할 때, 바코드 대신 RFID Tag의 ID값을 읽어 제품을 인식하는 태그이다[6].



(그림 2) EPCglobal Network 흐름 개념도

③ RFID Reader

RFID Reader는 RFID Tag에 들어가 있는 정보를 Air Interface를 통해 읽고, 쓰고, 비활성화 시키는 것과 같은 기능을 지원하는 역할을 수행한다. 즉, EPC 어플리케이션간의 데이터와 명령의 교환을 담당해 주는 역할을 한다.

④ ALE(Application Level Events)

ALE가 하는 역할은 RFID 태그를 인식한 리더기가 어플리케이션 계층으로 데이터를 전달하는 역할을 하는 일종의 미들웨어이다. 분산 컴퓨터 시스템에서 미들웨어는 오퍼레이팅 시스템과 어플리케이션 시스템 사이에 서로간의 데이터를 중개해 준다[5].

⑤ EPCIS(EPC Information Service)

EPCIS에서는 물리 객체의 고유한 성격에 대한 데이터인 정적 정보를 관리하는데 이는 변경되지 않는 클래스 레벨의 정보(예: 상품명, 상품코드, 제조사명 등)와 물리 객체마다 변경되는 인스턴스 레벨의 정보(예: 제조일자, 유통기한 등)를 말한다. 이를 통해 물리 객체의 동질성과 동시에 유일성을 제공한다. 또한 동적 정보(예: 상품의 SCM상의 위치정보, 입출고 및 판매 정보 등)는 추후 사용을 위해 로컬 저장소에 저장하고 관리한다[5].

⑥ ONS(Object Naming Service) 및 Local ONS

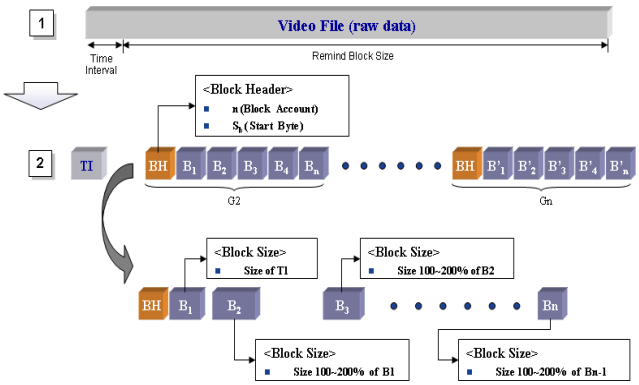
ONS는 인터넷의 검색방법인 DNS와 유사하게 정보를 검색한다. 즉 EPCglobal에서는 전세계에 광범위하게 퍼져있는 정보를 검색하고 인터넷의 네트워크를 사용하기 위해서 EPC라는 코드들을 DNS가 이해할 수 있도록 웹 서비스 시스템에 기초를 둔 시스템이다[5].

⑦ EPCIS Discovery Service

EPC Discovery Service는 EPCglobal Network에서 객체(EPC)의 추적을 위한 정보를 제공하는 서비스이다. 다시 말하면 EPCglobal Network 구성원이 특정 EPC에 대한 정보를 찾고 이를 접근할 수 있도록 요청하는 일련의 서비스를 의미한다.

이러한 EPCglobal Network를 제품 교환의 표준 시스템으로 사용하면 공급망의 가시성을 향상시켜 상품의 추적과 자동화를 높일 수 있으며, 상품 현황의 실시간 파악이 가능하여 제품 손실을 최소화 할 수 있으며 고객의 요구에 능동적으로 대처할 수 있다.

하지만 여전히 수많은 데이터의 저장과 전송량으로 인해 많은 트래픽을 부가한다. 따라 한 기업안에서도 EPCIS를 분리함으로써 보다 효율적인 시스템 구현을 하고 있으며, 아직까지 인증 암호화에 대한 스펙은 제시되지 않고 있으며,



(그림 3) 디지털 콘텐츠 블록 분할 및 암호화 방법

X.509 PKI 시스템을 이용할 것이라는 것만 암시해 두고 있는 실정이여서, 인증 및 암호화 방법에 대해 중요한 이슈로 남고 있다[7].

2.5 대칭키와 공개키를 함께 활용한 시스템

이번 절에서는 암호화 할 데이터를 대칭키로 암호화하고, 암호화에 사용된 암호화키는 공개키로 암호화 및 전자서명을 하여 사용자에게 안전하게 전달해 주는 시스템에 대해 설명하고자 한다. 이 연구는 멀티미디어 동영상 데이터를 (그림 3)과 같이 암호화하기 전에 원시데이터(Raw Data)를 블록으로 나누어 각각의 블록을 대칭키 암호화 알고리즘을 사용하여 서로 다른 대칭키를 이용하여 암호화 한다.

이와 같이 디지털콘텐츠에 사용된 암호화 키를 공개키 알고리즘으로 암호화하여 사용자에게 전송해주는 시스템이다. 디지털콘텐츠 분배 시스템에서 공개키만을 사용한 방법은 고용량의 멀티미디어 데이터를 실시간으로 미리 압축을 할 수 없을 뿐만 아니라 암호화 한다고 해도 서버의 오버헤드 문제로 인해 사용되지 못하고 있다. 따라서 디지털콘텐츠를 미리 대칭키로 암호화 시켜 놓은 후에 암호화에 사용된 암호화 키만을 공개키로 암호화하여 사용자에게 전송시켜 주는 방법을 사용하고 있다[8].

2.6 EPCglobal Network 메시지 교환시의 문제점

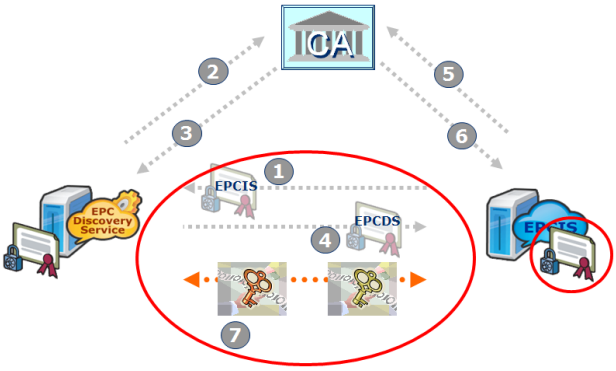
EPCglobal Architecture Framework에서 제안하는 x.509 기반의 공개키 암호화 방법은 대칭키 암호화 방법에 비해 암복호화 속도가 느린 반면, 암복호화시 서로 다른 키를 사용함으로 인해 안전성 측면에서는 효율적이다. 하지만 EPCIS나 ALE의 경우 실시간 처리를 하는 시스템으로서 단순히 공개키 알고리즘을 사용하여 메시지의 암복호화를 사용할 때 2.3절의 대칭키와 공개키를 함께 활용한 시스템에서 알 수 있듯이 오버헤드가 많이 발생한다. 이를 해결하고자 본 논문에서는 응답데이터는 대칭키로 암호화하여 전송하고, 암호화데이터의 위변조를 막기 위해 전자서명을 한 해쉬값을 붙여 전송한다. 또한 여기에 사용된 암호화키만을 공개키로 암호화과정을 통해 시스템의 오버헤드를 줄여보고자 한다.

3. 제안 시스템

3.1 제안 시스템의 인증과정

(그림 4)에서는 EPCIS와 EPCDS간의 암호화 과정을 보여주고 있으며, 그 외에도 EPCIS와 EPCIS간의 암호화 시스템등 각각의 시스템에서도 같은 방법으로 암호화하여 전송한다.

① EPCIS 인증서를 전자서명을 통해 EPCDS로 전송
② EPCIS 인증서 상태 검증요청
③ EPCIS 인증서 상태검증(EPCDS에서는 CRL or OCSP 과정을 통해 EPCIS에 대한 인증확인)
④ EPCDS 인증서를 전자서명을 통해 EPCIS로 전송
⑤ EPCDS 인증서 상태 검증요청
⑥ EPCDS 인증서 상태검증(EPCIS에서는 CRL or OCSP 과정을 통해 EPCDS에 대한 인증확인, 양방향 인증 완료



(그림 4) 제안하는 시스템의 인증 및 암호화

서로간의 인증이 완료가 되면, 모든 데이터에 대해 각각의 시스템이 가지고 있는 개인키를 통해 전자서명 하여, 메시지의 무결성을 확인하게 되며, 인증서에 포함되어 있는 상대방의 공개키를 통해 암호화하여 데이터를 전달한다. EPCglobal Network에서의 데이터 전송은 기본적으로 SOAP (Simple Object Access Protocol)을 통해 전송이 된다. 따라 본 메시지 역시 모두 SOAP을 통해 전송이 된다.

이와는 다르게 Local ONS와 EPCIS간의 기업 내에 있는 시스템 인증은 이와는 다르게 다음과 같은 3가지 암호화 정책을 지원하도록 한다.

Username Over Transport : 알려진 클라이언트와의 통신시 SSL 인증서를 사용하는 서버와 메시지 인증 및 암호화시 사용

Username Over Certificate : 알려진 클라이언트와의 통신시 SSL 인증서 방법이 아닌 일반 PKI 인증방법을 이용한 서버에서의 메시지 암호화 및 인증시 사용

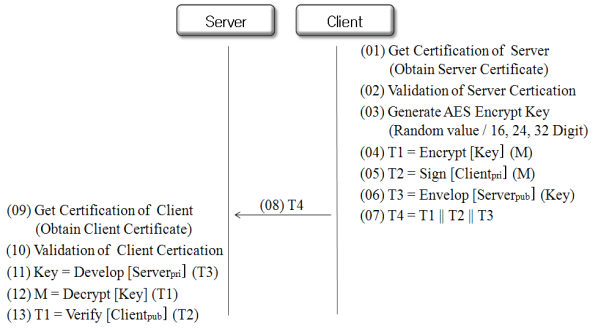
Kerberos : Windows 시스템간에 Active Directory와 같은 도메인으로 구성되어진 시스템간 인증시 사용

또한 인증서가 없는 Client에서 EPCIS의 데이터를 검색할 때는 Anonymous Over Certificate 암호화 정책을 사용한다. 이는 익명의 사용자가 EPCIS의 데이터를 검색할 시에만 사용한다.

3.2 메시지 암호화 과정

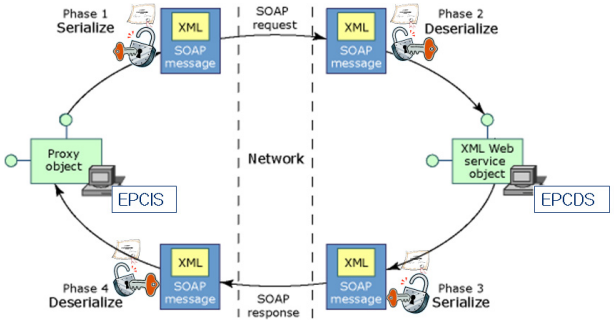
EPCglobal Network에서 각각의 기업간의 전송된 데이터를 통해 해당 데이터베이스에 Select, Insert, Update, Delete 간의 작업이 일어나게 되며, 만약 허위정보가 들어왔을 때에는 더 이상 데이터를 신용할 수 없는 데이터이기 때문에 시스템간의 인증과정과 중간의 공격자에 의해 전송한 메시지의 무결성을 검증해야 한다. 본 논문에서 제안하는 암호화 메시지를 통해 시스템간의 인증 및 무결성, 데이터 암호를 모두 할 수 있는 방법을 제시하며, 다음 (그림 5)와 같다.

제안하는 시스템에서 암호화 과정에서의 시스템은 위에서도 언급했듯이 EPCIS와 EPCDS간의 시스템이 아니며, 데이터를 먼저 요청하는 곳을 Client, 데이터를 전송받아 처리하는 곳을 Server라고 명한다. Client에서 Server에게 메시지를 전송하기 위해 13번의 단계를 거치며 각 단계별 프로세스는 다음과 같다.



(그림 5) 암호화 메시지 전송 방법

① Server의 인증서를 SOAP 메시지로 획득
② Server의 인증서 상태 검증 (CA에 요청)
③ AES 암호화 Key를 난수값으로 생성 (T1)
④ 전송할 메시지를 AES 암호화 키로 암호화 (T2)
⑤ 메시지를 Client의 개인키로 전자서명 (T3)
⑥ 단계 ①에서 획득한 Server의 인증서 안에 포함된 공개키를 이용하여 데이터 암호화
⑦ 단계 ④⑤⑥의 각각의 값들을 연접
⑧ 단계 ⑦의 값을 SOAP 메시지로 Server로 전송
⑨ Client의 인증서를 SOAP 메시지로 획득
⑩ Client의 인증서 상태 검증 (CA에게 요청)
⑪ Server의 개인키를 통해 T3의 메시지 복호화 (AES 암호화 키 획득)
⑫ AES 암호화 알고리즘을 가지고 메시지 T1 값을 단계 ⑪에서 획득한 암호화 키를 사용하여 복호화 (원문 메시지 획득)
⑬ 단계 ⑨에서 획득한 Client의 공개키를 통해 메시지의 위



(그림 6) 암호화 복호화 시점

변조가 없는지 전자 서명값 검사

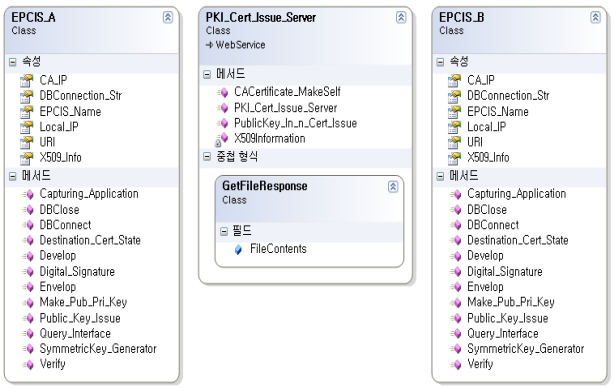
(그림 6)에서 암호화하는 시점과 복호화 하는 시점은 모든 데이터가 SOAP이라는 방법으로 서로 전송하기 때문에 암호화한 각각의 데이터를 직렬화(Serialize)를 통해 구성한 다음 MTOM(Message Transmission Optimization Mechanism)하여 전송을 하며, 복호화 하는 시점은 역직렬화(Deserialize) 과정을 통해 데이터를 얻은 후 복호화 하게 된다.

암호화 키가 서로 교환되어 있는 상태에서는 대칭키로만 암호화된 메시지를 보내주면 되며, 스푸핑 공격을 막기 위해서 메시지 뒤에 TimeStamp를 같이 붙여 전송을 한다.

4. 제안 시스템 구현

본 논문에서는 닷넷 기반의 CA, 클라이언트 및 서버를 구축하고 해당 작업의 요청을 위해 Web Services 방법으로 데이터를 전송 및 전달받게 된다.

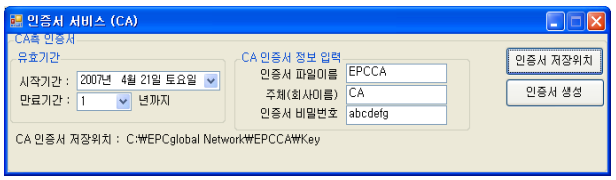
구현에 사용된 프로그램은 Microsoft Visual Studio 2005 C#, Microsoft Web Service Enhancements 3.0, Microsoft SQL Server 2005를 사용하였다.



(그림 7) 제안하는 시스템의 속성 및 메서드

4.1 CA의 인증서 발급 인터페이스

다음 (그림 9)은 EPCCA에서 개인키와 공개키, 그리고 자신의 키로 인증서를 생성하는 인터페이스이다.



(그림 8) CA 인터페이스

인증서를 생성하는 코드는 다음과 같다.

```
// CA의 개인키와 공개키 생성
nRet = Rsa.MakeKeys(CAPublicKeyFileName, CAPrivateKeyFileName, 1024, Rsa.PublicExponent.Ex p_EQ_65537, 1024, CACertPassword, Rsa.PbeOpti ons.Default, true);

//CA 인증서 생성
nRet = X509.MakeCertSelf(CACertFileName, CAPrivateKeyFileName, 1, CAExpireYear, "C=KR;O=EPCGlobal Network;OU=RetailTech;CN=" + CAName + "", "", X509.KeyUsageOptions.DigitalSignature | X509.KeyUsageOptions.KeyCertSign, CACertPassword, X509.Options.FormatPem);
```

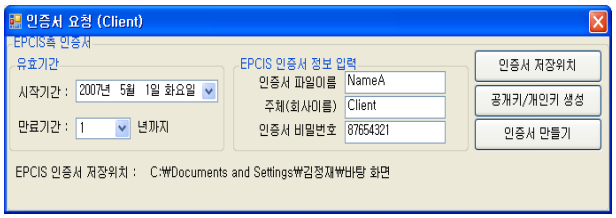
다음 코드는 다른 원격지의 컴퓨터에서 공개키를 보내주면 인증서로 생성하여 전송시켜 주는 EPCCA의 인증서 생성에 관한 코드이다.

```
// CA의 개인키와 공개키 생성
nRet = Rsa.MakeKeys(CAPublicKeyFileName, CAPrivateKeyFileName, 1024, Rsa.PublicExponent.Ex p_EQ_65537, 1024, CACertPassword, Rsa.PbeOpti ons.Default, true);

//CA 인증서 생성
nRet = X509.MakeCertSelf(CACertFileName, CAPrivateKeyFileName, 1, CAExpireYear, "C=KR;O=EPCGlobal Network;OU=RetailTech;CN=" + CAName + "", "", X509.KeyUsageOptions.DigitalSignature | X509.KeyUsageOptions.KeyCertSign, CACertPassword, X509.Options.FormatPem);
```

4.2 Client의 키생성 및 인증서 요청 인터페이스

다음 그림은 Client에서 개인키와 공개키를 생성하고 공개키를 인증서서버(EPCCA)의 웹서비스에 요청하게되면 EPCCA에서는 EPCCA 인증서와 개인키를 가지고 전자서명한 후, Client 인증서를 생성하여 다시 Client로 전송해주게 된다.



(그림 9) Client의 인증서 요청 인터페이스

이때 CA에서 발급된 인증서를 클라이언트에게 전송할 때는 MTOM 메시지로 전송하게 된다.

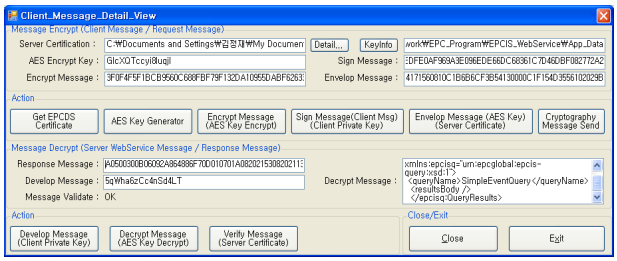
다음 그림은 EPCCA에서 Client로 전송하는 SOAP 메시지의 일부분이다.

```
- <soap:Body>
  - <CertMakeFromPubKeyResponse xmlns="http://tempuri.org/">
    - <getFileResponse
        xmlns="http://locahost/WebService/EPCCA">
        <fileName>Client.Cer</fileName>

          <fileData>MIICEjCCAXsCAWYwDQYJKoZIhvcNAQEEBQA
      </getFileResponse>
    </CertMakeFromPubKeyResponse>
  </soap:Body>
```

(그림 10) 인증서 요청 SOAP 메시지

다음 (그림 11)은 Client 메시지를 Server에게 전송하고 그 결과값을 다시 Client가 다시 복호화하는 인터페이스이다.



(그림 11) Client 암호화 인터페이스

4.3 서버에서의 SOAP 메시지

다음 그림은 Client에서 Server로부터 메시지를 전송해 주는 SOAP 메시지이다.

<AES_Encrypt_Msg>의 엘리먼트는 본 논문의 3.2 Message 암호화 전송과정에서의 T1에 해당하는 메시지이며, 원문 메시지가 AES 암호화 알고리즘으로 암호화 된 메시지이다. <Sign_Msg> 엘리먼트는 T2에 해당하는 메시지이며, 원문 메시지를 해쉬한 값으로 원문메시지를 복호화 하였을 때, 위변조가 없는지 확인하기 위한 값이다. <Envelop_Key> 엘리먼트는 T3에 해당하는 메시지로, Server의 공개키를 가지고 암호화 시킨 값으로 구성되어 있다.

```
- <soap:Body>
  - <EncryptMsgResponse xmlns="http://tempuri.org/">
      <SystemName>Client</SystemName>

        <AES_Encrypt_Msg>C4228209C2DD16E397A98C78A10560

        <Sign_Msg>3082034D06092A864886F70D010702A082033

        <Envelop_Key>424B18D9B1444E224C2B710CC69112FE5C
    </EncryptMsgResponse>
  </soap:Body>
```

(그림 12) Server가 전송받은 SOAP 메시지

Server에서 Client로 SOAP 메시지를 전송하기 위해서 T1, T2, T3 메시지를 복호화 하여 처리한 후, 다시 새로운 T1, T2, T3 메시지를 생성하여 Client로 보내주게 된다.

```
- <soap:Body>
  - <EncryptMsgResponseResponse
      xmlns="http://tempuri.org/">
    - <getEncryptResponse
        xmlns="http://localhost/WebService/Server">

          <AES_Encrypt_Data>V70J9LyXc39zgqjf8ZNNA4Y2pbv5G9gIGwz

          <EPCDS_Sign_Data>MIIDWgYJKoZIhvcNAQcCoIIDSzCCA0cCAQE

          <Envelop_Data>Cay3IQrAGVghDgf1i8szQSqvfp3copAMePjr0+u
      </getEncryptResponse>
    </EncryptMsgResponseResponse>
  </soap:Body>
```

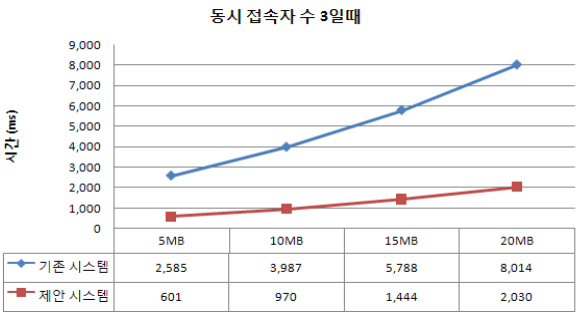
(그림 13) Client로 전송할 SOAP 메시지

5. 실험 평가

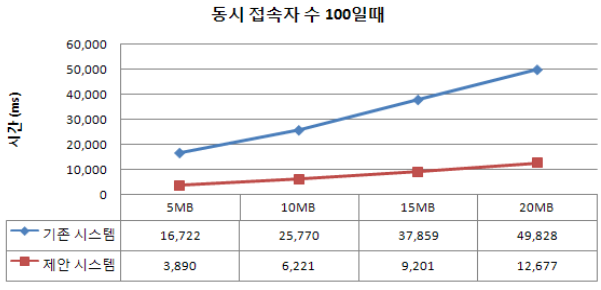
5.1 암복호화에 대한 시간 평가

본 논문에서 사용되는 실험 결과값은 Request XML 문서는 40KByte로 설정하였으며, Server 쪽의 Response 데이터는 5Mbyte, 10Mbyte, 15Myte, 20MByte XML 데이터를 가지고 실험평가를 하였다. 또한 다량의 접속자가 동시에 데이터를 처리하는 환경으로 처리하기 위해 Web Application Stress Tool을 사용하였다. 동시 접속자 수를 3~100 User로 설정하여 메시지가 처리완료 되기까지의 시간을 측정하였다.

기존의 알려진 대칭키 알고리즘을 사용하지 않고 공개키 알고리즘만을 사용한 방법과 제안하는 시스템에서의 암복호화 시간을 비교하였으며, 클라이언트로부터 받은 메시지를 복호화하여, 결과값을 처리하고, 다시 암호화하여 클라이언트로 전송하기 전까지의 시간을 비교하였다.



(그림 14) 동시 접속자 수가 3일 때의 처리



(그림 15) 동시 접속자 수가 100일 때의 처리

위의 실험은 서버에서만 측정한 값이며, 제안한 시스템이 공개키만을 사용한 방법보다 약 4배 정도 향상되었으며, 이는 대칭키 알고리즘이 공개키 알고리즘보다 암호화 및 복호화 속도가 빠르기 때문이며, 시스템의 오버헤드가 감소됨을 의미한다. 하지만 제안하는 시스템의 경우 Client에서 Server로 전송되는 데이터 중 암호화 메시지(XML 메시지)를 제외한 값에는 기존의 시스템보다 약 2Kb 정도 증가하였는데 증가한 이유는 전자 서명에서의 해쉬값과 공개키로 암호화된 대칭키를 전송했기 때문이다.

5.2 메시지 안전성에 대한 평가

이번 절에서는 암호화 메시지에 대한 안전성을 평가한다.

- 스니핑에 대한 공격

암호화 메시지는 각 시스템의 개인키가 유출되지 않으면 암호화 메시지를 복원할 수 없다. 아직 공개키 알고리즘에 대한 취약점이 발표되지 않았으며, 안전성을 높이기 위해 현재 표준인 1024Bit를 사용하였다.

- 스푸핑에 대한 공격

스푸핑 공격을 막기 위해 본 논문에서는 메시지 순번에 대한 TimeStamp를 사용하여 보안성을 증가시켰다. 만약 데이터를 가로챈 후에 값을 변조 시키더라도 개인키 없이는 변조가 불가능하기 때문이다. 또한 공개키 및 대칭키 암호화 알고리즘에 사용된 암호화 프로그램은 NIST에서 실시하고 있는 CMVP(Cryptographic Module Validation Program)에서 안전하다고 평가되었으며, SOAP의 MTOM 메시지로 전달함으로 인해 Direct Internet Message Encapsulation (DIME) 방식인 첨부 파일의 보안시에도 전송 수준의 보안 또는 메시지 수준의 보안을 쉽게 사용할 수 있도록 다른 종류의 자격 정보의 지원(커버로스 암호화, 사용자 정책(Policy)) 등 메시지 수준의 보안을 추가만 하여 사용할 수 있는 장점이 있다.

<표 1>은 제안한 시스템에 대한 장·단점을 설명하였다.

〈표 1〉 제안시스템의 장·단점

	장 점	단 점
기 존 시스템	-	• 암·복호화 속도 느림 • 오버헤드 증가 가능성 존재
제 안 시스템	• 기존시스템보다 암·복호화 속도 증가 • 오버헤드 감소 • 스니핑 / 스푸핑에 강함	• Client->Server 메시지 전송시 약 2Kb 증가 (전자서명 해쉬값과 대칭키 전송으로 인한 증가)

6. 결 론

본 논문에서는 EPCglobal Network상에서 각 시스템간의 데이터 통신시에 안전하게 메시지를 보낼 수 있도록 할 수 있는 암호화 시스템을 제안 및 구현하였다. 이는 EPCglobal Network에서 Spec으로 나오는 X.509의 기본 원칙을 그대로 지켰을 뿐 아니라 메시지 복호화시 대량의 데이터를 속도 향상을 위해 대칭키를 사용하여 암호화 하였으며, 여기에 사용된 키만을 개인키로 암호화 시켰으며, 데이터의 무결성을 위해 메시지를 전자서명을 통해 수정하는 것을 방지하였다. 또한 시스템간의 인증을 하기 위해 인증서를 통해 서로 인증할 수 있도록 하였다. 향후 ALE와 Reader, ONS 등을 포함한 실제 EPCglobal Architecture Framework을 구현하여 암호화 성능을 측정할 것이다.

참 고 문 헌

[1] EPCglobal, http://www.epcglobalinc.org, 2008.
[2] http://www.nist.gov
[3] John Linn, “Trust Models and Management in Public Key Infrastructures,” Technical Notes and Reports of RSA Laboratories, November, 2000.
[4] EPCglabal network기반의 RFID 기술 및 활용, 글로벌.
[5] The EPCglobal Architecture Framework, EPCglobal Final Version 1.2 Approved 10, September, 2007.
[6] Understanding the EPC Gen 2 Protocol, RFID Journal Special Report, 2005.
[7] EPCglobal Certificate Profile, Ratified Specification 1.0, March, 8, 2006.
[8] 김정재 외 3명, “DRM 시스템에서 해쉬체인과 세션키 교환을 이용한 암호화 기법에 관한 연구”, 한국정보처리학회 논문지 C, Vol.13-C No. 07 pp.0843-0850, 2006. 12.

김 대 중
e-mail : djkim@fsb.or.kr
2000년 8월 한국방송통신대학교 컴퓨터과학과(이학사)
2004년 2월 숭실대학교 업기술정보대학원 정보통신공학과(공학석사)
2008년 8월 숭실대학교 일반대학원 컴퓨터학과(공학박사)
2008년 11월 상호저축은행중앙회 IT본부 전산기획팀장
관심분야 : 보안, 정보통신, 암호학, RFID

김 정 재
e-mail : argniss@yahoo.co.kr
1999년 2월 영동대학교 컴퓨터공학과(공학사)
2001년 2월 숭실대학교 컴퓨터학과(공학석사)
2005년 8월 숭실대학교 컴퓨터학과(공학박사)
2006년 7월 (주)레일테크 기술연구소 수석연구원
관심분야 : DRM, 암호학, RFID

이 승 민

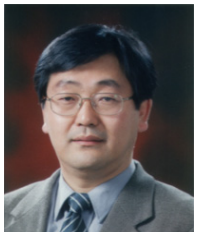
e-mail : cowaboonga@hotmail.com

2004년 2월 한서대학교 컴퓨터정보학과 (이학사)

2006년 2월 숭실대학교 일반대학원 컴퓨터학과(공학석사)

2007년 2월 숭실대학교 일반대학원 컴퓨터학과 박사과정

관심분야 : 보안, 정보통신, RFID, 네트워크, PKI



전 문 석

e-mail : mjun@computing.ssu.ac.kr

1981년 숭실대학교 전자계산학과(공학사)

1986년 University of Maryland Computer Science(공학석사)

1989년 University of Maryland Computer Science(공학박사)

1989년 3월~7월 Morgan State University 조교수

1989년~1991년 New Mexico State University Physical Science Lab 책임 연구원

1991년~현 재 숭실대학교 컴퓨터학부 부교수

관심분야 : 전자상거래 보안, 인터넷 보안, 멀티미디어 보안, 인증시스템, PKI, RFID